Decomposition and Mixed Integer Programs

Laszlo Ladanyi
IBM T.J. Watson Research Center

David Jensen
IBM T.J. Watson Research Center

Outline

- Application: FCC Auction #31
- Solving IPs with branch-and-bound using an unusual relaxation
- Incorporating cutting planes to create branch-cut-price
- Treating a secondary objective via complementarity
- Application: FCC Auction #31 (revisited)
- Extending the algorithm to general MIP

Application: FCC Auction #31 – 1

Wireless frequency licenses are auctioned off.

- Iterative auction: repeat until no more new bids
  - Bid submission: regulated by complex rules (eligibility, bid survival, etc.) See Public Notices.
  - Bid evaluation: given the bids, compute a “fair” revenue-maximizing provisional allocation of licences.
- Bids may be submitted for individual licences or for bundles of licences.

Application: FCC Auction #31 – 2

- primary objective: maximize revenue
- secondary objective: random (ensures fairness: a random choice between alternate optima)
- Target: bid evaluation and feedback computation in less than 15 mins
- These are IP's that *must* be solved to optimality
- Major reservation against package bidding was its computational complexity

Optimization at the end of a round

Bid evaluation

- Stage I: Select a revenue-maximizing subset of bids
  - consider bids from all rounds so far
  - XOR of OR bids: bidder may win any bids from a round but all his winning bids must come from the same round
- Stage II: Select one of the optimal solutions randomly
  - achieved by optimizing wrt random secondary objective
  - traditionally implemented by adding the primary objective as a constraint

Stage I: Revenue maximization

For each agent $a \in A$ and round $t \in T$ define:

- $M_{a,t}$: the matrix whose columns are the incidence vectors of bids
- $\mathbf{v}_{a,t}$: the array of objective coefficients corresponding to these bids
- $\mathbf{x}_{a,t}$: binary variables indicating which of these bids are accepted
- $y_{a,t}$: a binary variable indicating whether any of these bids are selected or not.

Stage I: disaggregated formulation

objective:
$$\min \sum_{a,t} [\mathbf{v}_{a,t}^T, 0] \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix}$$

license constraints
$$\sum_{a,t} [M_{a,t}, \mathbf{0}] \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} \leq \mathbf{1}$$

bidder constraints
$$\sum_{t} [\mathbf{0}^T, 1] \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} \leq 1 \qquad \forall a$$

bid-round constraints
$$[M_{a,t}, -\mathbf{1}] \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} \leq \mathbf{0} \qquad \forall a, t$$

$$\mathbf{x}_{a,t}, y_{a,t} \in \{0, 1\} \qquad \forall a, t$$

Column Generation Reformulation – 1

Formulated by Dietrich & Forrest:

- Variables correspond to *proposals*: possible bid combinations of a bidder. The vector of variables for bidder a is $\boldsymbol{\lambda}_a$.
- Formulation of master problem
  - List proposals of bidder a in X_a
  - Require that at most one proposal per bidder is selected
  - Require that $\boldsymbol{\lambda}_a$'s are integral
- Subproblems used to dynamically generate proposals

Column Generation Reformulation – 2

Master Problem	Subproblems for each a,t
$\min \sum_a \mathbf{v}_a^T X_a \boldsymbol{\lambda}_a$	$\min \left(\mathbf{v}_{a,t}^T, -M_{a,t}^T \boldsymbol{\pi}\right) \mathbf{x}_{a,t} - \nu_a$
$\sum_a X_a \boldsymbol{\lambda}_a \leq \mathbf{1}$	$M_{a,t}\mathbf{x}_{a,t} \leq \mathbf{1}$
$\mathbf{e}^T \boldsymbol{\lambda}_a \leq \mathbf{1} \quad \forall a$	$\mathbf{1} \geq \mathbf{x}_{a,t} \geq \mathbf{0}$
$\boldsymbol{\lambda}_a \in \{0,1\} \quad \forall a$	$\mathbf{x}_{a,t}$ binary

Solve via branch-and-bound.

General IP problem considered

$$
\begin{aligned}
& \min \sum_{i=1}^{k} \mathbf{c}_i^T \mathbf{x}_i \\
(IP) \qquad & \sum_{i=1}^{k} A_i \mathbf{x}_i \leq \mathbf{b} \\
& D_i \mathbf{x}_i \leq \mathbf{d}_i \qquad \forall i = 1, ..., k \\
& \mathbf{x}_i \text{ binary} \qquad \forall i = 1, ..., k
\end{aligned}
$$

- “hard” connecting constraints
- block-diagonal “easy” constraints
- binary requirement just for easier notation, trivial to relax to real MIP

Solving (IP) with Branch-and-Bound

Branching: any combination of changing bounds on constraints and/or variables (just to simplify discussion; easy to generalize)

Bounding: Solve the bounding via Dantzig-Wolfe decomposition

Dantzig-Wolfe for bounding

Original relaxation (the $\mathbf{b}'$, $\mathbf{d}'_i$, and $\mathbf{l}_i$, $\mathbf{u}_i$ vectors reflect the branching decisions)

$$\begin{aligned} \min \textstyle\sum_{i=1}^{k} \mathbf{c}_i^T \mathbf{x}_i & \\ \textstyle\sum_{i=1}^{k} A_i \mathbf{x}_i \le \mathbf{b}' & \\ D_i \mathbf{x}_i \le \mathbf{d}'_i & \quad \forall i \end{aligned} \qquad \text{such that} \qquad \begin{aligned} D_i \mathbf{x}_i \le \mathbf{d}'_i & \qquad \forall i \\ \mathbf{l}_i \le \mathbf{x}_i \le \mathbf{u}_i & \qquad \forall i \end{aligned}$$

Dantzig-Wolfe decomposition:

Master Problem

$$\begin{aligned} \min \textstyle\sum_{i=1}^{k} \mathbf{c}_i^T X_i \boldsymbol{\lambda}_i & \\ \textstyle\sum_{i=1}^{k} A_i X_i \boldsymbol{\lambda}_i \le \mathbf{b}' & \\ \mathbf{e}^T \boldsymbol{\lambda}_i = 1 & \quad \forall i \\ \boldsymbol{\lambda}_i \ge \mathbf{0} & \quad \forall i \end{aligned}$$

The i$^{\text{th}}$ subproblem

$$\begin{aligned} \min(\mathbf{c}_i^T - A_i^T \boldsymbol{\pi})\mathbf{x}_i - \boldsymbol{\delta}_i & \\ D_i \mathbf{x}_i \le \mathbf{d}'_i & \\ \mathbf{l}_i \le \mathbf{x}_i \le \mathbf{u}_i & \end{aligned}$$

- $\boldsymbol{\pi}$: the dual vector corresponding to the “hard” constraints
- δ_i: the dual value corresponding to the i^{th} convexity constraint.

Dantzig-Wolfe for bounding – a tightened version

Original relaxation (the $\mathbf{b}'$, $\mathbf{d}'_i$, and $\mathbf{l}_i$, $\mathbf{u}_i$ vectors reflect the branching decisions)

$$\begin{aligned} \min \textstyle\sum_{i=1}^{k} \mathbf{c}_i^T \mathbf{x}_i & \\ \textstyle\sum_{i=1}^{k} A_i \mathbf{x}_i \le \mathbf{b}' & \\ D_i \mathbf{x}_i \le \mathbf{d}'_i & \quad \forall i \end{aligned} \qquad \text{such that} \qquad \begin{aligned} D_i \mathbf{x}_i \le \mathbf{d}'_i & \quad \forall i \\ \mathbf{l}_i \le \mathbf{x}_i \le \mathbf{u}_i & \quad \forall i \\ \mathbf{x}_i \text{ integer} & \quad \forall i \end{aligned}$$

Dantzig-Wolfe decomposition:

Master Problem

$$\begin{aligned} \min \textstyle\sum_{i=1}^{k} \mathbf{c}_i^T X_i \boldsymbol{\lambda}_i & \\ \textstyle\sum_{i=1}^{k} A_i X_i \boldsymbol{\lambda}_i \le \mathbf{b}' & \\ \mathbf{e}^T \boldsymbol{\lambda}_i = 1 & \quad \forall i \\ \boldsymbol{\lambda}_i \ge \mathbf{0} & \quad \forall i \end{aligned}$$

The i$^{\mathrm{th}}$ subproblem

$$\begin{aligned} \min (\mathbf{c}_i^T - A_i^T \boldsymbol{\pi}) \mathbf{x}_i - \boldsymbol{\delta}_i & \\ D_i \mathbf{x}_i \le \mathbf{d}'_i & \\ \mathbf{l}_i \le \mathbf{x}_i \le \mathbf{u}_i & \\ \mathbf{x}_i \text{ integer} & \end{aligned}$$

Relaxation can be significantly tighter; depends on the integrality gap in the subproblems.

Further tightening the relaxation: cut generation

For a solution $(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k)$ to the Master Problem $(X_1\boldsymbol{\lambda}_1, \ldots, X_k\boldsymbol{\lambda}_k)$ is a solution to the original.

$\Rightarrow$ generate cuts $\sum_{i=1}^{k} F_i\mathbf{x}_i \leq \mathbf{f}$ in the original space.

$\Rightarrow$ "Incorporate" F into A, i.e., add constraints $\sum_{i=1}^{k} F_i X_i \boldsymbol{\lambda}_i \leq \mathbf{f}$ to the Master Problem.

The duals of the new constraints are incorporated into the objectives of the subproblems.

Note: there might be violated cuts for the master problem (in the traditional branch-and-price sense, i.e., when $\boldsymbol{\lambda}_i$ is assumed to be integer) while there are none for the original problem.

(Generalized) branching, real MIP

- Branching done in original space, e.g., bound changes according to the integrality of $\mathbf{x}_i = X_i \boldsymbol{\lambda}_i$. Such changes are directly moved into the subproblems.
- Can generalize branching from “change bounds” to “branching on general hyperplanes”, i.e., “add cuts and change bounds”. Additional cuts are incorporated into A, the set of “hard” constraints.
- General MIP properties, i.e., general bounds on the variables and allowing continuous variables trivially carry over to the subproblems.

End result: branch-cut-price

- Original formulation is never explicitly maintained
- $\Rightarrow$ in effect branch-cut-price is implemented on the master problem where integrality of $X_i \boldsymbol{\lambda}_i$ is required

- In traditional branch-and-price integrality of $\boldsymbol{\lambda}_i$ is required
- $\Rightarrow$ hence the trouble with cut generation (the duals of cuts generated for the master problem can't be interpreted)

Is it worth?

Con:

- Subproblems are IPs. Dantzig-Wolfe decomposition is slow to converge to begin with, how slow it will be now?

Pro:

- Subproblems are IPs.
  - the larger the integrality gap in the subproblems the tighter the relaxation and the better the algorithm
  - Unlike in DW for LP, here the column set can be seeded by solving the original LP and applying heuristics to get solutions to the IP subproblems.
- Excellently parallelizable
  - branch and cut can reasonably process only a few dozen search tree nodes in parallel
  - with decomposition many processors can be used for one node
  - scales up to BlueGene size parallelism.

Lexicographic optimization

After optimizing wrt. a primary objective (Stage I.) we need to further optimize wrt. a secondary objective (Stage II.):

$$
\begin{array}{lll}
 & \min \sum_{i=1}^{k} \mathbf{v}_i^T \mathbf{x}_i & \\
(IP-2) & \sum_{i=1}^{k} A_i \mathbf{x}_i \leq \mathbf{b} & \\
 & D_i \mathbf{x}_i \leq \mathbf{d}_i & \forall i = 1, ..., k \\
 & \mathbf{x}_i \text{ binary} & \forall i = 1, ..., k \\
 & \mathbf{x}_i \text{ minimizes primary objective} &
\end{array}
$$

- Traditionally done by adding an extra constraint
- $\Rightarrow$ degeneracy, numerical instability.

Alternative solution: complementarity for Stage II.

Idea: stay on the optimal face by enforcing complementarity.

- Explore Stage I. search tree.
- Discard leaves with lower bound > optimal primary value.
- In the rest of the leaves find alternate optimal solution with best secondary objective value and take best of those:
  - Suppose all subproblems solved as LP when D-W terminated;
  - $\Rightarrow$ the leaf might as well have been bounded via LP relaxation;
  - $\Rightarrow$ can create dual optimal solution to original formulation;
  - $\Rightarrow$ can use complementarity to fix bounds to stay on LP optimal face;
  - $\Rightarrow$ primary objective will not change, can continue branch and bound with secondary objective.

Exploiting complementarity

Let $\boldsymbol{\pi}$ be the dual vector in the master problem and $\boldsymbol{\gamma}_i$'s be the dual vectors of the subproblems. Then $(\boldsymbol{\pi}, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_k)$ is dual optimal to the original formulation.

- if (in the original formulation) the reduced cost $c_i^j - \boldsymbol{\pi}^T A_i^j - \boldsymbol{\gamma}_i^T D_i^j$ of variable x_i^j is negative (positive) then the variable must be fixed at its current upper (lower) bound for Stage II.
- if the dual value π^k is negative (positive) then the k^{th} row of the original problem (and the master problem) must be fixed at its current upper (lower) bound for Stage II.
- if the dual value γ_i^k is negative (positive) then the k^{th} row of the i^{th} subproblem must be fixed at its current upper (lower) bound for Stage II.

Removing the "solve as LP" assumption

When D-W terminates, for each subproblem that does not solve as an LP do *NOT* carry over the subproblem to Stage II, rather:

- Explore the search tree of the subproblem.
- Concentrate on the leaves where lower bound = optimal value
- For all such leaves
  - create a subproblem in Stage II. with the appropriate bound changes that define this leaf;
  - however, these subproblems will share the convexity constraint of the original subproblem.

FCC Auction #31: Stage I

objective: $$\min \sum_{a,t} [\mathbf{v}_{a,t}^T, 0] \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix}$$

license constraints $$\sum_{a,t} [M_{a,t}, \mathbf{0}] \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} \leq \mathbf{1}$$

bidder constraints $$\sum_{t} [\mathbf{0}^T, 1] \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} \leq 1 \qquad \forall a$$

bid-round constraints $$[M_{a,t}, -\mathbf{1}] \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} \leq \mathbf{0} \qquad \forall a, t$$

$$\mathbf{x}_{a,t}, y_{a,t} \in \{0, 1\} \qquad \forall a, t$$

- apply Branch-and-Bound to this formulation
- bounding at search tree nodes is via Dantzig-Wolfe (bid-round + binary are "easy")

FCC Auction #31: Dantzig-Wolfe

Applying Dantzig-Wolfe to lower bounding (license round constraints and x, y binary are "easy"):

$$
\begin{aligned}
\min \sum_{a,t} \left[\mathbf{v}_{a,t}^T, 0\right] \begin{bmatrix} X_{a,t} \\ \mathbf{y}_{a,t} \end{bmatrix} \boldsymbol{\lambda}_{a,t} & \\
\sum_{a,t} \left[M_{a,t}, \mathbf{0}\right] \begin{bmatrix} X_{a,t} \\ \mathbf{y}_{a,t} \end{bmatrix} \boldsymbol{\lambda}_{a,t} &\leq \mathbf{1} \\
\sum_{t} \left[\mathbf{0}^T, 1\right] \begin{bmatrix} X_{a,t} \\ \mathbf{y}_{a,t} \end{bmatrix} \boldsymbol{\lambda}_{a,t} &\leq \mathbf{1} \quad \forall a \\
\mathbf{e}^T \boldsymbol{\lambda}_{a,t} &= \mathbf{1} \quad \forall a, t \\
\boldsymbol{\lambda}_{a,t} &\geq \mathbf{0}
\end{aligned}
$$

$$
\begin{aligned}
\min \left(\left[\mathbf{v}_{a,t}^T, 0\right] - \left[M_{a,t}^T, \mathbf{0}\right] \boldsymbol{\pi} - \left[\mathbf{0}^T, 1\right] \nu_a \right) \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} - \delta_{a,t} & \\
\left[M_{a,t}, -\mathbf{1}\right] \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} &\leq \mathbf{0} \\
\mathbf{1} \geq \begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} &\geq \mathbf{0} \\
\begin{bmatrix} \mathbf{x}_{a,t} \\ y_{a,t} \end{bmatrix} & \text{ binary}
\end{aligned}
$$

- replace $=$ with $\leq$ in convexity constraints (0 is solution to subproblem)
- Claim: throughout column generation $\delta_{a,t}$ will always be 0.
  - Proof: the bidder constraints dominate the convexity constraints hence there is an optimal solution to the master problem with all δ's being 0.
- In master problem discard convexity constraints (they'll be always dominated by the bidder constraints)
- In subproblems set $y_{a,t}$ to 1 (when it is 0 the problem is rather uninteresting).

FCC Auction #31: resulting formulation

Master Problem

$$\min \sum_{a,t} \mathbf{v}_{a,t}^T X_{a,t} \boldsymbol{\lambda}_{a,t}$$

$$\sum_{a,t} M_{a,t} X_{a,t} \boldsymbol{\lambda}_{a,t} \leq \mathbf{1}$$

$$\sum_{t} \mathbf{e}^T \boldsymbol{\lambda}_{a,t} \leq \mathbf{1} \quad \forall a$$

$$\boldsymbol{\lambda}_{a,t} \geq \mathbf{0} \quad \forall a, t$$

Subproblems

$$\min \left(\mathbf{v}_{a,t}^T, -M_{a,t}^T \boldsymbol{\pi}\right) \mathbf{x}_{a,t} - \nu_a$$

$$M_{a,t} \mathbf{x}_{a,t} \leq \mathbf{1}$$

$$\mathbf{1} \geq \mathbf{x}_{a,t} \geq \mathbf{0}$$

$\mathbf{x}_{a,t}$ binary

Note: was non-trivial to eliminate the y variables.

- Identical to the formulation of Dietrich and Forrest.
- intuitive column generation same as Dantzig-Wolfe based

FCC Auction #31: Implementation and results

- branching on license: whether or not a license is assigned to a particular bidder. Easily enforced in Master Problem and Subproblems.
- generated clique and odd hole inequalities

- Stage I. computation is fast (the subproblems usually solve as LPs) Stage II. is instantenous, in effect the problem is fixed.
  - 12 licences, up to 44 rounds, 6-7000 bids, up to 30 bidders (20-30 instances) under 2 seconds
  - 50 licences, 15K bids, 16 rounds, 50 bidders (5 instances) about 2.5 minutes; second stage never takes more than a couple of seconds – this is usually the difficult stage.
  - 150 licences, 10K bids, 50 bidders, 4 rounds (1 instance) about 20 minutes; second stage no more than a couple of seconds.

- Implementation used the BCP framework and the Cut Generation Library from http://www.coin-or.org

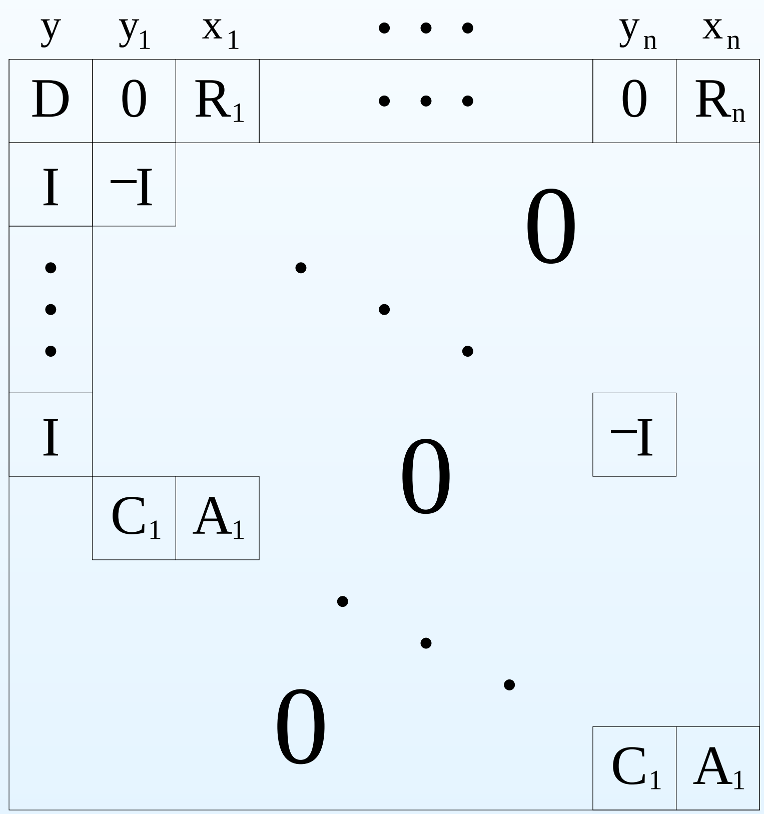
General MIP

- current algorithm works when matrix is Dantzig-Wolfe decomposable (block diagonal with connecting constraints)
- what if there are connecting variables as well?

$$\begin{array}{cccc} y & x_1 & \cdots & x_n \\ \hline D & R_1 & \cdots & R_n \\ C_1 & A_1 & & 0 \\ \vdots & & \ddots & \\ C_n & 0 & & A_n \end{array}$$

Transform to decomposable MIP

- Introduce variables $y_i = y$ for all i
- $\Rightarrow$ Dantzig-Wolfe decomposable



Computational results

None... Every problem we looked at is non-decomposable, Dantzig-Wolfe decomposable or Benders decomposable.

Actively soliciting problems...